

**CLAIMS**

1           1.     A method for dynamically patching code, comprising the steps of:  
2           intercepting program instructions;  
3           determining if a program instruction requires unavailable hardware  
4           functionality; and  
5           dynamically replacing the program instruction with a replacement instruction  
6           that does not require unavailable hardware functionality if it is determined that the  
7           program instruction requires unavailable hardware functionality.

1           2.     The method of claim 1, wherein the step of dynamically replacing the  
2           program instruction comprises fetching a replacement instruction and storing it in a  
3           code cache.

1           3.     The method of claim 2, wherein the step of dynamically replacing the  
2           program instruction further comprises executing the replacement instruction in lieu of  
3           the program instruction each time a function associated with the program instruction  
4           is required.

1           4.     The method of claim 3, wherein the replacement instruction comprises  
2           part of a patch that is made available via an application programming interface.

1           5.     The method of claim 1, further comprising the step of, prior to  
2           determining if a program instruction requires unavailable hardware functionality,  
3           determining if the program instruction has been cached.

1           6.       The method of claim 5, further comprising the step of executing the  
2       cached instruction in lieu of the program instruction if an associated instruction has  
3       been cached.

1           7.       The method of claim 1, further comprising the step of, prior to  
2       intercepting program instructions, gaining control over execution of program  
3       instructions by injecting a dynamic execution layer interface into the program.

1           8.       The method of claim 1, further comprising the step of dynamically  
2       receiving information about unavailable hardware functionality and replacement  
3       instructions that are configured to replace original program instructions that require  
4       the unavailable hardware functionality.

1           9.       A system for dynamically patching code, comprising:  
2               means for gaining control over execution of a program;  
3               means for intercepting program instructions;  
4               means for determining if a program instruction requires unavailable hardware  
5       functionality; and  
6               means for dynamically replacing the program instruction with a replacement  
7       instruction that does not require unavailable hardware functionality if it is determined  
8       that the program instruction requires unavailable hardware functionality.

1           10.      The system of claim 9, wherein the means for dynamically replacing  
2       the program instruction comprise means for fetching a replacement instruction and  
3       storing it in a code cache.

1           11.     The system of claim 9, further comprising means for determining if a  
2     program instruction has been cached.

1           12.     The system of claim 9, further comprising means for dynamically  
2     receiving information about unavailable hardware functionality and replacement  
3     instructions that are configured to replace original program instructions that require  
4     the unavailable hardware functionality.

1           13.     A dynamic patching program stored on a computer-readable medium,  
2     comprising:

3           logic configured to gain control over execution of a program;

4           logic configured to intercept program instructions;

5           logic configured to determine if a program instruction requires unavailable  
6     hardware functionality; and

7           logic configured to dynamically replace the program instruction with a  
8     replacement instruction that does not require unavailable hardware functionality if it  
9     is determined that the program instruction requires unavailable hardware  
10    functionality.

1           14.     The system of claim 13, wherein the logic configured to dynamically  
2     replace the program instruction comprises logic configured to fetch a replacement  
3     instruction and store it in a code cache.

1           15.     The system of claim 13, further comprising logic configured to  
2     determine if a program instruction has been cached.

1           16.    The system of claim 13, further comprising logic configured to  
2   dynamically receive information about unavailable hardware functionality and  
3   replacement instructions that are configured to replace original program instructions  
4   that require the unavailable hardware functionality.

1           17.    A method for dynamically patching code, comprising the steps of:  
2           gaining control over the execution of a program;  
3           intercepting program instructions;  
4           determining whether the program instructions have been cached and, if so,  
5   executing the cached instructions;  
6           if the program instructions have not been cached, determining if the program  
7   instructions require unavailable hardware functionality; and  
8           dynamically replacing the program instructions with replacement instructions  
9   that do not require unavailable hardware functionality if it is determined that the  
10   program instructions require unavailable hardware functionality.

1           18.    The method of claim 17, wherein the step of dynamically replacing the  
2   program instructions comprises fetching replacement instructions and storing them in  
3   a code cache.

1           19.    The method of claim 18, wherein the step of dynamically replacing the  
2   program instructions further comprises executing the replacement instructions in lieu  
3   of the program instructions each time a functionality associated with the program  
4   instructions is required.

1           20.    The method of claim 19, wherein the replacement instructions  
2   comprise part of a patch that is made available via an application programming  
3   interface.

1           21.    A dynamic execution layer interface (DELI) residing between an  
2   application and computing system hardware, comprising:

3           a transparent mode layer that is configured to gain control over the operation  
4   of the application and to fetch replacement instructions that are to replace existing  
5   application instructions;

6           a system control and configuration layer configured to provide policies for the  
7   replacement of existing application instructions with the replacement instructions;

8           a core configured to dynamically cache and execute the replacement  
9   instructions; and

10          a code cache in which the replacement instructions are cached.

1           22.    The DELI of claim 21, wherein the transparent mode layer is further  
2   configured to fetch application instructions from the application and wherein the core  
3   is further configured to cache fetched application instructions in the code cache.

099575-112901  
FO627T-524660